

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Лабораторна робота № 7

з дисципліни «Алгоритмізація та програмування»

Тема: "Реалізація алгоритмів обробки двовимірних масивів на мові C ++"

Виконала студент гр. 319а

Хара Дмитро

(підпис, дата) (П.І.Б.)

Перевірив

(підпис, дата)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів (матриць) на мові C++ і реалізувати оголошення, введення з консолі, обробку і виведення в консоль матриць на мові C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці. Введення і виведення даних здійснити в командному вікні. Варіанти завдань представлено в табл.1.

Matrix43. Дана матриця розміру $M \times N$. Знайти кількість її стовпців, елементи яких впорядковані за спаданням.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання (див. табл.2.), розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

Matrix68. Дана матриця розміру $M \times N$ і ціле число K ($1 \leq K \leq M$). Перед рядком матриці з номером K вставити рядок з нулів.

Хід роботи

Код програми

```
#include <iostream>
#include <Windows.h>
using namespace std;
void matrix43();
void matrix68();

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int menu;
    do
    {
        cout << "Номер завдання(0 для виходу):";
        cin >> menu;
        switch (menu)
        { // перемикання між завданнями
            case 1:
                matrix43();
                break; // Завдання 1
            case 2:
                matrix68();
                break; // Завдання 2
        }
    } while (menu);
    system("pause");
    return 0;
}

void input_matrix(int* matrix[], int rows, int cols)
{
    cout << "Введіть елементи матриці:\n";
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            cout << "Елемент [" << i << "][" << j << "]: ";
            cin >> matrix[i][j];
        }
    }
}
```

```

    }
}

// функція виведення масиву
void print_matrix(int* matrix[], int rows, int cols) {
    cout << "Матриця розміру " << rows << " x " << cols << ":" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

int countColumnsDescending(int** matrix, int rows, int cols) {
    int count = 0;

    for (int j = 0; j < cols; ++j) {
        bool descending = true;
        for (int i = 1; i < rows; ++i) {
            if (matrix[i][j] > matrix[i - 1][j]) {
                descending = false;
                break;
            }
        }
        if (descending) {
            count++;
        }
    }

    return count;
}

void matrix43()
{
    int M, N;

    // Зчитуємо розміри матриці
    cout << "Введіть кількість рядків (M): ";
    cin >> M;
    cout << "Введіть кількість стовпців (N): ";
    cin >> N;

    // Створюємо матрицю розміру M x N
    int** matrix = new int* [M];
    for (int i = 0; i < M; ++i) {
        matrix[i] = new int[N];
    }
    input_matrix(matrix, M, N);
    // Знаходимо кількість стовпців з елементами, що впорядковані за
    спаданням
    int count = countColumnsDescending(matrix, M, N);
    cout << "Кількість стовпців з елементами, що впорядковані за спаданням:
" << count << endl;

    // Звільняємо виділену пам'ять
    for (int i = 0; i < M; ++i) {
        delete[] matrix[i];
    }
    delete[] matrix;
}

void insertZeroRow(int**& matrix, int& M, int N, int K) {

```

```

// Зберігаємо початковий розмір матриці
int originalM = M;

// Збільшуємо розмір матриці на один рядок
M++;
// Переносимо рядки від K до M на один рядок вниз
for (int i = M - 1; i >= K; --i) {
    matrix[i] = matrix[i - 1];
}

// Вставляємо рядок з нулів перед рядком з номером K
matrix[K - 1] = new int[N]();
}
void matrix68()
{
    int M, N, K;

    // Зчитуємо розміри матриці та номер рядка K
    cout << "Введіть кількість рядків (M): ";
    cin >> M;
    cout << "Введіть кількість стовпців (N): ";
    cin >> N;
    cout << "Введіть номер рядка для вставки ( $1 \leq K \leq M$ ): ";
    cin >> K;

    int** matrix = new int* [M];
    for (int i = 0; i < M; ++i) {
        matrix[i] = new int[N];
    }
    // Введення та виведення матриці
    input_matrix(matrix, M, N);
    print_matrix(matrix, M, N);

    // Викликаємо функцію для вставки рядка з нулів перед рядком з номером
K
    insertZeroRow(matrix, M, N, K);
    print_matrix(matrix, M, N);
}

```

```
Номер завдання(0 для виходу):1
Введіть кількість рядків (M): 4
Введіть кількість стовпців (N): 3
Введіть елементи матриці:
Елемент [0][0]: 5
Елемент [0][1]: 4
Елемент [0][2]: 3
Елемент [1][0]: 4
Елемент [1][1]: 3
Елемент [1][2]: 2
Елемент [2][0]: 3
Елемент [2][1]: 2
Елемент [2][2]: 1
Елемент [3][0]: 2
Елемент [3][1]: 1
Елемент [3][2]: 0
Кількість стовпців з елементами, що впорядковані за спаданням: 3
Номер завдання(0 для виходу):2
Введіть кількість рядків (M): 3
Введіть кількість стовпців (N): 3
Введіть номер рядка для вставки (1 ? K ? M): 2
Введіть елементи матриці:
Елемент [0][0]: 1
Елемент [0][1]: 2
Елемент [0][2]: 3
Елемент [1][0]: 4
Елемент [1][1]: 5
Елемент [1][2]: 6
Елемент [2][0]: 7
Елемент [2][1]: 8
Елемент [2][2]: 9
Матриця розміру 3 x 3:
1 2 3
4 5 6
7 8 9
Матриця розміру 4 x 3:
1 2 3
0 0 0
4 5 6
7 8 9
```